Linear multi-objective drift analysis

Rowe, Jonathan E.

DOI:
10.1016/j.tcs.2018.02.011

Linear multi-objective drift analysis

Jonathan E. Rowe

School of Computer Science,
University of Birmingham, Birmingham, B15 2TT,
United Kingdom
J.E.Rowe@cs.bham.ac.uk

Abstract

The tools of drift analysis enable bounds on run-times (or first hitting times) of stochastic processes, such as randomised algorithms, based on bounds on the expected progress at each time step in terms of a distance measure. In this paper, we generalise the *multiplicative drift* theorem to apply to processes which are best described by more than one distance function. We provide four examples to illustrate the application of this method: the run-time analysis of an evolutionary algorithm on a multi-objective optimisation problem; the analysis of a variant of the *voter* model on a network; a parallel evolutionary algorithm taking place on islands with limited migration; a synchronous network epidemiology model. In the latter example, we show that populations with limited neighbourhoods (such as the ring topology) are able to resist epidemics much more effectively than well-mixed populations.

Keywords: run-time analysis, multiplicative drift analysis, multi-objective optimisation, evolutionary algorithms, voter model, network epidemiology

1. Introduction

Drift analysis has now become a standard tool in the analysis of randomised search heuristics (see, for example, [9, 10]). The fundamental idea is that, given an estimate of the expected progress towards some target at each time step, we can use this to bound the expected time for the process to reach that target. Upper and lower bound versions exist. We will consider the case of *multiplicative drift* [2], and will generalise this to the situation where the process concerned is described by more than one distance measure.

We begin, then, by considering a finite state space $\mathcal{X}$ and a random process $X_0, X_1, X_2, \ldots$ on this set. Suppose we are interested in the first hitting time, T, of a target set $S \subseteq \mathcal{X}$. We describe the progress towards this target set by means of a *distance function*.

Definition 1. *Given a set $\mathcal{X}$ and a target set $S \subseteq \mathcal{X}$ a* distance function*, with respect to S, is a function $d : \mathcal{X} \to \mathbb{R}$ such that:*

$x \in S \Longrightarrow d(x) = 0$ *and*

$x \notin S \Longrightarrow d(x) > 0$.

It should be noted that we do not necessarily expect a distance function to be a metric.

We estimate the progress of the random sequence towards S by bounding the expected change in the distance. This then allows us to bound the expected first hitting time of the target.

Theorem 1 (Multiplicative drift [1, 2]). *Let $X_0, X_1, X_2, \ldots$ be a random sequence from the finite set $\mathcal{X}$. Let $S \subseteq \mathcal{X}$ be a target set, and let $d : \mathcal{X} \to \mathbb{R}$ be a distance function with respect to S. Let T be the first hitting time of the target set S.*

Suppose, for all states $x \notin S$, we have

$$E[d(X_{t+1}) \,|\, X_t = x] \leq \delta d(x)$$

for some constant $0 < \delta < 1$. Then:

$$E[T \,|\, X_0 = x_0] \leq \frac{1 + \log(d(x_0)/d^*)}{1 - \delta}$$

where

$$d^* = \min\{d(x) \,|\, x \notin S\}.$$

The probability that the first hitting time exceeds

$$\frac{\log(d(x_0)/d^*) + c}{1 - \delta}$$

for any $c > 0$ is no more than

$$e^{-c}.$$

It should be noted that, while there is a sharp tail bound on the probability that the expected hitting time exceeds the upper bound, the bound can, in some situations, be rather conservative. A simple example that illustrates this is the *unexploded bomb* problem, in which there are n bombs that, at a given time step, each have probability p of exploding. If there are currently k unexploded bombs, then the expected number of unexploded bombs at the next time step is $(1 - p)k$. The multiplicative drift theorem them tells us that the expected time until they have all exploded is bounded above by $(1 + \log n)/p$. This is quite a good estimate when p is small, but as p approaches 1, the expected time actually approaches 1 and not $1 + \log n$. The upper bounds are better when the process described typically only makes small jumps.

Our goal in this paper is to extend the multiplicative drift theorem to the case where the underlying random sequence is best described by more than one distance function, and where the progress in each distance function considered separately is not necessarily monotonic. We will, in particular, look at the

situation where the expected change in each distance function is a linear function of all the distances, and provide conditions under which first hitting time bounds can be proved. This turns out to be relatively straightforward for the case of two distance functions. Dealing with a larger number of functions requires sufficient structure in the problem to make progress. However, it is possible, in some situations, to deal even with an arbitrary number of dimensions. Our result will inherit the strengths (in terms of the sharp tail bound) and the weaknesses (in terms of the conservatism) of the original multiplicative drift theorem.

We will illustrate our result with four examples. Firstly, we will look at proving the run-time of a simple evolutionary algorithm on a multi-objective optimisation problem. Then we will look at the so-called *voter* model on a graph, examining a variant in which each voter has an inherent preference. A generalisation of that model allows us to analyse an evolutionary process taking place in parallel on multiple islands, with migration between neighbouring islands. Finally, we look at a synchronous model of epidemiology on a network and derive conditions for which epidemics will fail to take hold in the population. In particular, we will show that a population with very limited neighbourhood structure (we will consider a ring topology) is far better able to resist an epidemic than a well-mixed population.

2. Multi-objective drift

Now suppose our process is described by several distance functions $d_1, \ldots, d_m$ such that the intersection of the corresponding target sets is not empty. This intersection is now our new target set S. The expected change in each of these distances can depend on each other as follows. Let A be a non-negative $m \times m$ matrix. Then we suppose that, for each d_i and $x \notin S$, we have

$$E[d_i(X_{t+1}) \mid X_t = x] \leq \sum_j A_{i,j} d_j(x).$$

In order to find a measure of our overall progress, we will define a new distance function, which will be a convex combination of $d_1, \ldots, d_m$.

Suppose that A has a left eigenvector $\mathbf{v}$ which contains only real, positive entries. We can assume that this eigenvector is normalised so that its components sum to 1. Let λ be the corresponding eigenvalue.

$$\mathbf{v}A = \lambda \mathbf{v}.$$

We will use the normalised $\mathbf{v}$ to define our new distance function. Letting $\mathbf{d}(x)$ be the (column) vector $(d_1(x), \ldots, d_m(x))$, our new distance function is simply $\mathbf{v} \cdot \mathbf{d}(x)$.

Theorem 2 (Linear multi-objective drift). *Let $X_0, X_1, X_2, \ldots$ be a random sequence from the finite set $\mathcal{X}$. Let $d_1, \ldots, d_m$ be distance functions with respect to target sets $S_1, \ldots, S_m$ respectively, and suppose $S = \bigcap S_i$ is non-empty.*

Writing $\mathbf{d}(x) = (d_1(x), \ldots, d_m(x))$ *as a column vector, suppose there is a non-negative matrix* A *such that, for all* $x \notin S$*:*

$$E[\mathbf{d}(X_{t+1}) \,|\, X_t = x] \leq A\mathbf{d}(x)$$

component-wise. If A *has a left eigenvector* $\mathbf{v}$ *which contains only real, positive entries, and which has a corresponding real eigenvalue* $0 < \lambda < 1$*, then the expectation of the first hitting time* T *of the set* S *(that is, the set of states for which all distances are zero) is bounded above by:*

$$E[T \,|\, X_0 = x_0] \leq \frac{1 + \log(\mathbf{v} \cdot \mathbf{d}(x_0)/d^*)}{1 - \lambda}$$

where

$$d^* = \min\{\mathbf{v} \cdot \mathbf{d}(x) \,|\, x \notin S\}.$$

Proof
We let $\mathbf{v} \cdot \mathbf{d}(x)$ be a new distance function. Since components of $\mathbf{v}$ are positive, this function is zero only when all of $d_1, \ldots, d_m$ are zero. Then

$$\begin{aligned} E[\mathbf{v} \cdot \mathbf{d}(X_{t+1}) \,|\, X_t = x] &= \mathbf{v} \cdot E[\mathbf{d}(X_{t+1}) \,|\, X_t = x] \\ &\leq \mathbf{v} \cdot (A\mathbf{d}(x)) \\ &= (\mathbf{v}A) \cdot \mathbf{d}(x) \\ &= \lambda \mathbf{v} \cdot \mathbf{d}(x). \end{aligned}$$

Since $0 < \lambda < 1$, this means we can apply the multiplicative drift theorem to our new distance function. □

We remark that the sharp tails bounds of Theorem 1 apply directly: the probability that the run-time exceeds the upper bound is vanishingly small.

In many applications, distances are integers in some bounded range. In this case, we can get the following simpler bounds:

Corollary 3. *If for all* $x \notin S$ *we have* $1 \leq d_i(x) \leq n$ *for each distance function* $d_1, \ldots, d_m$ *then*

$$E[T \,|\, X_0] \leq \frac{1 + \log(n/d^*)}{1 - \lambda}$$

where

$$d^* = \min\{\mathbf{v} \cdot \mathbf{d}(x) \,|\, x \notin S\}.$$

Noting that, in this situation, d^* *must be at least the value of the smallest component of* $\mathbf{v}$*, we also have:*

$$E[T \,|\, X_0] \leq \frac{1 + \log(n/v^*)}{1 - \lambda}$$

where

$$v^* = \min_i v_i.$$

In the special case that the matrix A is irredicible, the Perron-Frobenius theorem tells us immediately that there is an eigenvector $\mathbf{v}$ with all real, positive components. The corresponding eigenvalue is real and positive, and has the largest absolute value of all the eigenvalues of A. It is therefore equal to the spectral radius of the matrix $\rho(A)$. Moreover, any eigenvector with all positive components is associated with this eigenvalue.

When there are only two distance functions, the spectral radius can be written down explicitly. This enables us to derive a simple check that the linear multi-objective drift theorem is applicable.

Lemma 4. *Let A be an irredicuble, non-negative 2×2 matrix*

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}.$$

Then $\rho(A) < 1$ if and only if

$$(1 - a_{1,1})(1 - a_{2,2}) > a_{1,2}a_{2,1}.$$

Proof
The eigenvalues of A satisfy

$$\begin{vmatrix} a_{1,1} - \lambda & a_{1,2} \\ a_{2,1} & a_{2,2} - \lambda \end{vmatrix} = 0$$

so the spectral radius is the larger of the two roots:

$$\lambda = \frac{a_{1,1} + a_{2,2} + \sqrt{(a_{1,1} + a_{2,2})^2 - 4(a_{1,1}a_{2,2} - a_{1,2}a_{2,1})}}{2}.$$

Solving for $\lambda < 1$ gives the required condition. □

3. Example — multi-objective optimisation

Suppose we have the following multi-objective problem on $\{0,1\}^n$.
Minimise:

$f_1(x) = \textsc{OneMax}(x) = \sum_{i=1}^{n} x_i$

$f_2(x) = \textsc{Diff}(x) = \sum_{i=1}^{n} (1 - x_i)x_{n-i+1}$.

The first objective is the standard test function which counts the number of ones in a string. The second measures the extent to which the string is a palindrome: all palindromic strings have optimal fitness for this objective. We use a simple multi-objective version of the Random Local Search (RLS) algorithm (see Algorithm 1).

Algorithm 1 Multi-objective Random Local Search

Let x be chosen randomly from $\{0,1\}^n$.
while true **do**
    Pick $i \in \{1,\dots n\}$ uniformly at random.
    Let y be x with bit i flipped.
    If either $f_1(y) < f_1(x)$ or $f_2(y) < f_2(x)$ then let $x = y$.
end while

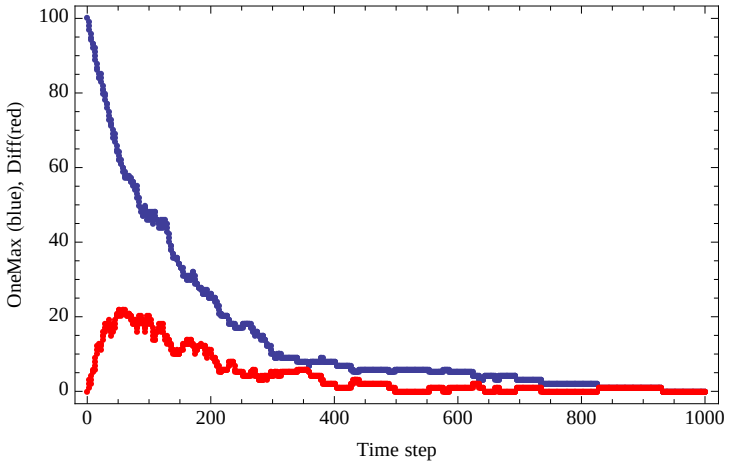

Figure 1: A typical run for the (OneMax,Diff) problem, with $n = 100$, starting from the all ones string. Upper line is OneMax, lower line is Diff.

The target set is the single string containing only zeros. Notice, however, that improvement in second objective (increasing the number of matches) can make the first objective worse. Similarly, removing a one will improve the first objective but might make the second worse. Looking at a typical run (Figure 1), starting with the all ones string, we see that the number of ones generally goes down, whereas the second objective, which starts at an optimum, gets progressively worse for a while before a general improvement takes place.

If X_t is the state at time t, let $\zeta_i(X_t)$ be the random variable representing the value of bit i at timestep t. Then

$$E[\zeta_i(X_{t+1}) \,|\, X_t = x] = \left(1 - \frac{1}{n}\right) x_i + \frac{1}{n}(1 - x_i)x_{n-i+1}$$

since, for bit i to be a one at step $t+1$ it is either already a one and is not changed, or is it a zero at step t, gets mutated, and the mutation is accepted because the matching bit at position $n - i + 1$ is a one.

Summing gives our first distance function:

$$E[\textsc{OneMax}(X_{t+1}) \,|\, X_t = x] = \left(1 - \frac{1}{n}\right) \textsc{OneMax}(x) + \frac{1}{n}\textsc{Diff}(x).$$

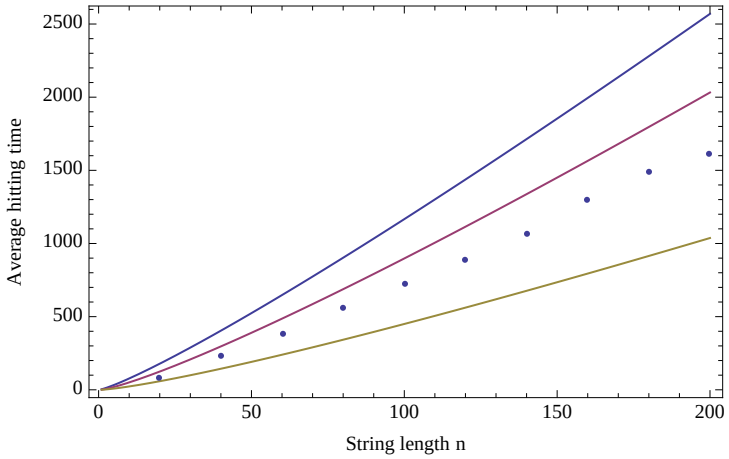

Figure 2: The average time to the all zero string for the (OneMax,Diff) problem, with random initial string. Each data point is averaged over 100 runs. Upper bounds from Corollary and Theorem. Lower bound from OneMax as a single objective.

Now for each bit position i, let

$$\chi_i(x) = [x_i = 0 \text{ and } x_{n-i+1} = 1] = (1 - x_i)x_{n-i+1}.$$

Then

$$E[\chi_i(X_{t+1}) \mid X_t = x] = (1 - 2/n)\chi_i(x) + (1/n)[x_i = 1 \text{ and } x_{n-i+1} = 1]$$

since either the mismatch already exists and is not mutated (either mutations would be accepted) or we have a pair of ones and bit i is flipped (which again would be accepted). There are two other possible configurations. If $x_i = x_{n-i+1} = 0$ then no mutation of these bits would be accepted. And if $x_i = 1$ and $x_{n-i+1} = 0$ then while either mutation would be successful, neither would create a configuration satisfying χ_i.

Summing gives:

$$E[\textsc{Diff}(X_{t+1}) \mid X_t = x] = (1 - 2/n)\textsc{Diff}(x) + (\textsc{OneMax}(x) - \textsc{Diff}(x))/n$$

since the number of one-one pairs is given by the total number of ones, less those for which there is a mismatch. Thus:

$$E[\textsc{Diff}(X_{t+1}) \mid X_t = x] = \frac{1}{n}\textsc{OneMax}(x) + \left(1 - \frac{3}{n}\right)\textsc{Diff}(x)$$

which is our second distance function.

We now have:

$$A = \begin{pmatrix} 1 - 1/n & 1/n \\ 1/n & 1 - 3/n \end{pmatrix}$$

which is non-negative and irreducible if $n > 3$, giving

$$\rho(A) = 1 - \frac{2-\sqrt{2}}{n}$$

and

$$\mathbf{v} = \left(\frac{\sqrt{2}}{2}, 1 - \frac{\sqrt{2}}{2}\right).$$

A simple application of the Corollary tells us:

$$E[T \mid X_0] \leq \frac{n(\log n + \log(2+\sqrt{2}) + 1)}{2-\sqrt{2}}.$$

However, we can do better than this by a more careful analysis of the new potential function. When there are k ones in the string, the DIFF function is maximised when the ones come first, followed by the zeros. In the case $k > n/2$ we have $\textsc{Diff}(x) = n - k$ and

$$\mathbf{v} \cdot (k, n-k) = (1 - \sqrt{2}/2)n + (\sqrt{2} - 1)k$$

which is maximised when $k = n$, for which $\mathbf{v} \cdot (k, n-k) = n/\sqrt{2}$. On the other hand, if $k <= n/2$, we have $\textsc{Diff}(x) = k$ and

$$\mathbf{v} \cdot (k, k) = k$$

which is maximised when $k = n/2$. We conclude that the potential function can never exceed $n/\sqrt{2}$. To find v^*, consider when the number of ones, k is even (but not zero). Then the smallest value that $\textsc{Diff}(x)$ can take is zero, and

$$\mathbf{v} \cdot (k, 0) = k/\sqrt{2}$$

which is smallest when $k = 2$. For odd values of k however, $\textsc{Diff}(x)$ can achieve a minimum of 1, and then

$$\mathbf{v} \cdot (k, 1) = k/\sqrt{2} + 1 - \sqrt{2}/2$$

which attains a value of 1 when $k = 1$. We can therefore set $v^* = 1$. Our theorem then gives us the run-time bound:

$$E[T \mid X_0] \leq \frac{n(\log n - \log\sqrt{2} + 1)}{2-\sqrt{2}}.$$

Some experimental runs illustrate this run-time in Figure 2. The starting configurations are randomly chosen strings, and the experiments are averaged over 100 runs. It can be seen that the bound deriving from the Corollary is somewhat weaker than the more careful analysis. The run-time of the RLS algorithm on ONEMAX is given as a comparison.

4. Example — voter model with inherent preference

In the classic voter model [11], nodes in a network can be in one of two states (0 or 1). At each time step a node will pick a random neighbour, and then change its state to match that of the neighbour. In the synchronous version, all nodes perform this update simultaneously.

We consider a variant of this model, in which there is an inherent preference for one of the states (state 0). That is, with probability $0 < \mu < 1$ a node will copy its state from a random neighbour as before. Otherwise, it will change its state to 0, with probability p, if it is not already in that state, and with probability $1 - p$ it will remain in its current state. The system will clearly be absorbed in the state where all nodes are in state 0, and we seek the expected time for this to happen. We allow the existence of isolated nodes (i.e. with no neighbours). For such nodes, with probability μ they will simply retain their current state (as they have no neighbours to influence them). Failing that, the state will update to 0 with probability p as with all other nodes.

If there are m nodes, then the stochastic process takes place over the set $\{0,1\}^m$. We let $d_i(x)$ represent the state of node i in configuration x; ν_i is the degree of node i; $i \sim j$ indicates that nodes i and j are neighbours; and I is the set of isolated nodes. Then

$$E[d_i(X_{t+1}) \mid X_t = x] = \begin{cases} (1-\mu)(1-p)d_i(x) + \frac{\mu}{\nu_i}\sum_{j\sim i} d_j(x) & \text{if } i \notin I, \\ ((1-\mu)(1-p)+\mu)d_i(x) & \text{if } i \in I, \end{cases}$$

which gives us:

$$A_{i,j} = \begin{cases} (1-\mu)(1-p) & \text{if } i = j \text{ and } i \notin I, \\ (1-\mu)(1-p)+\mu & \text{if } i = j \text{ and } i \in I, \\ \frac{\mu}{\nu_i}[i \sim j] & \text{if } i \neq j \text{ and } i \notin I, \\ 0 & \text{if } i \neq j \text{ and } i \in I. \end{cases}$$

Let

$$w_i = \begin{cases} \nu_i & \text{if } i \notin I, \\ 1 & \text{if } i \in I. \end{cases}$$

It can be checked that $\mathbf{w} = (w_1, \ldots, w_m)$ is a left eigenvector of A, with corresponding eigenvalue $\lambda = (1-\mu)(1-p)+\mu$.

We normalise the eigenvector to get

$$\mathbf{v} = \frac{1}{2|E| + |I|}\mathbf{w}$$

where E is the edge set of the network. Applying the linear multi-objective drift theorem gives the upper bound on the run-time as:

$$E[T \mid X_0] \leq \frac{1 + \log(2|E| + |I|) - \log(\min_i w_i)}{p(1-\mu)}.$$

We illustrate this result with a set of experiments on the star graph topology. Mindful of the fact that the multiplicative drift theorem gives tighter bounds

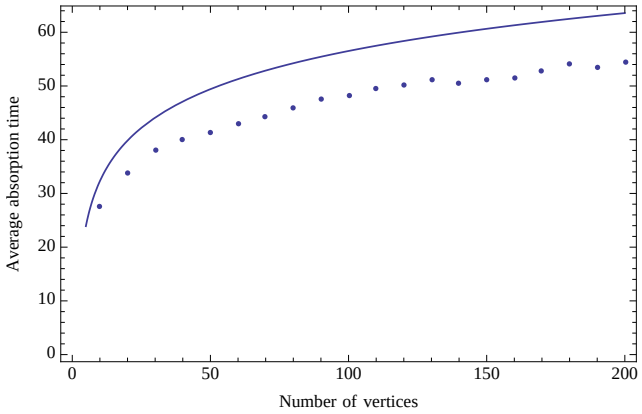

Figure 3: The average time for all nodes to reach state 0 for the voter model with preference on the star topology. Migration rate $\mu = 0.01$, local progress rate $p = 0.1$. Experiments averaged over 1000 runs. Solid line gives theoretical upper bound.

when progress is made in small steps, we first consider the case where the migration is low ($\mu = 0.01$) and the local progress is also low ($p = 0.1$). Figure 3 shows the time to absorption in this case, for increasing graph size.

The *unexploded bomb* example described above shows that if large steps towards the target occur, then the mulitplicative drift theorem can give a very conservative upper bound. We can reproduce this effect in the current model by increasing the local progress rate to $p = 0.99$. Figure 4 shows the result of doing this for different star graph sizes. It can be seen that the time to absorption is close to a constant, and the theoretical upper bound considerably overestimates this.

What about the effect of increasing the migration rate μ? This is interesting in revealing a second reason why the multiplicative drift theorem can be too conservative. Imagine the state of the star graph where the central node has state 1 and the others have state 0. If μ is high, and p is low, then in the next time step, the central node will be 0 and most of the others will be in state 1. The situation will reverse again at the next time step. This oscillation has the effect of making the variance in the run-time extremely large. Now, recall that the multiplicative drift theorem has a sharp tail bound: the probability that a run takes longer than the stated time is very small. Consequently, for situtations where the run-time exhibits a large variance, the upper bound given must necessarily be conservative. This is shown in Figure 5 where we have set $\mu = 0.9$ and $p = 0.1$. It is worth noting that the standard deviations of the run-times (not plotted) are approximately the same size as the means.

We now turn to a different example, which explores the effect of the dependence of the run-time result on the minimum vertex degree. Consider a complete graph on n vertices. The theoretical upper bound for the run-time is

$$\frac{1 + \log n}{p(1 - \mu)}$$

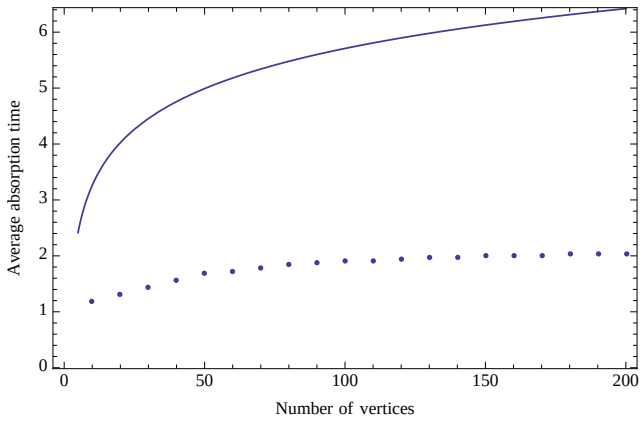

Figure 4: The average time for all nodes to reach state 0 for the voter model with preference on the star toplogy. Migration rate $\mu = 0.01$, local progress rate $p = 0.99$. Experiments averaged over 1000 runs. Solid line gives theoretical upper bound.

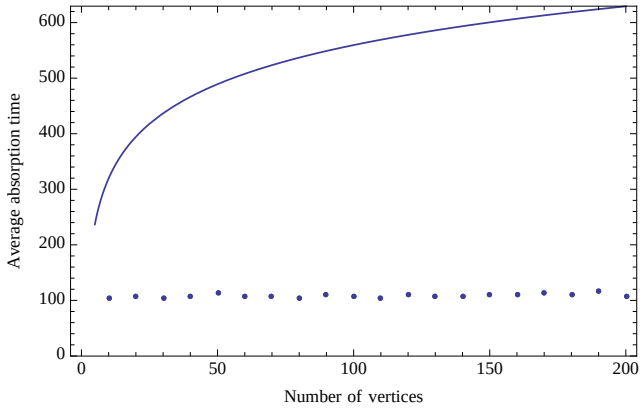

Figure 5: The average time for all nodes to reach state 0 for the voter model with preference on the star toplogy. Migration rate $\mu = 0.9$, local progress rate $p = 0.1$. Experiments averaged over 1000 runs. Solid line gives theoretical upper bound.

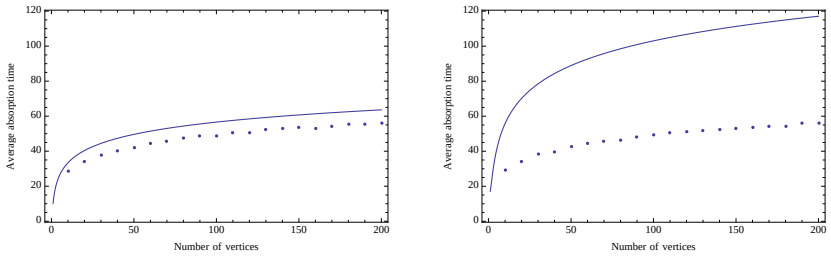

Figure 6: Time to absoprtion for the voter model with preference for the complete graph (left) and the complete graph augmented with a single vertex (right). Migration rate $\mu = 0.01$, local progress rate $p = 0.1$. Experiments averaged over 1000 runs. Solid line gives theoretical upper bound.

If we now add a vertex, and connect it to just one of the existing vertices, the theoretical upper bound becomes

$$\frac{1 + \log(n^2 - n + 2)}{p(1 - \mu)}$$

which is approximately double the original bound. It is hard to imagine that the addition of this single vertex could make such a difference in practice. Figure 6 shows that, indeed, the run-times of the two systems are almost identical. The effect on the upper bound is due to the fact that the minimum vertex degree is now 1 instead of $n - 1$.

The issue here in fact derives from a third factor that can cause the multiplicative drift theorem to over-estimate the run-time. The run-time bound is dependent on the distance of the intial state from the target divided by the distance of the state closest to the target. If it happens, as here, that visiting this state is rather unlikely, then the bound is going to be correspondingly conservative.

We have seen, in this section, three possible causes of how the multiplicative drift theorem can result in a large over-estimate of the run-time:

1. When large steps towards the target can occur.
2. When the variance in the run-time is large.
3. When the state closest to the target is unlikely to be visited.

The linear multi-objective drift theorem presented in this paper naturally inherits all these issues.

5. Example — evolution on parallel islands

One can generalise the previous example by considering the process that takes place at each node, when there is no interaction with neighbours, to be any process for which the multiplicative drift theorem holds, for some δ. Again, with probability μ, the state is copied from a random neighbour (or is unchanged, in

the case of an isolated node), otherwise the local process updates one time step. We therefore have

$$E[d_i(X_{t+1}) \mid X_t = x] \leq \begin{cases} (1-\mu)\delta d_i(x) + \frac{\mu}{\nu_i}\sum_{j\sim i} d_j(x) & \text{if } i \notin I, \\ (1-\mu)\delta d_i(x) + \mu d_i(x) & \text{if } i \in I. \end{cases}$$

The previous example has $\delta = 1 - p$.

So now consider an evolutionary process taking place in parallel on a number of islands [7]. This process takes place on strings from $\{0,1\}^n$, and follows the $(1+1)$EA on OneMax (see, for example, [3]). To be precise, the state at each node (island) is a binary string of length n. At each time step, with probability μ a random neighbour is chosen (if there are any), and the current state is replaced by a copy of the neighbour's string (or left unchanged, in the case of an isolated node). With probability $1-\mu$, however, a copy of the current string is made, and each bit is mutated with probability $1/n$. If the resulting string has fewer ones than the original, then it replaces it; otherwise the current string remains.

Eventually, all islands will hold the string containing all zeros — but how long will this take?

Let the distance functions be the number of ones in the current string at each island. For a single step of the $(1+1)$EA, consider the situation where the current string has k ones. The chance to mutate exactly one of them is

$$\left(1 - \frac{1}{n}\right)^{n-1} \frac{k}{n} \geq \frac{k}{en}$$

and so the expected next state is bounded above by $(1 - 1/en)k$. Adapting the analysis of the previous section, and using $\delta = 1 - 1/en$, gives a run-time bounded above by:

$$\frac{en(\log n + \log(2|E| + |I|) - \log(w^*) + 1)}{1-\mu}.$$

where w^* is the minimum vertex degree, if there are no isolated nodes, or 1 otherwise.

Experimental results in Figure 7 illustrate this result by considering copies the $(1+1)$EA optimising OneMax on n bits, running on each node of a ring topology with $m = n$ islands. Migration is set relatively low ($\mu = 0.1$). The considerations of high variance run-times, discussed in the previous section, would also apply here.

6. Example — network epidemiology

Models of the spread of infectious diseases usually assume that members of a population are in a particular state at a given time, for example being *infected* with the disease or *susceptible* to catching the disease (see, for example, [14]). It is assumed that infected individuals recover according to some rate. Susceptible

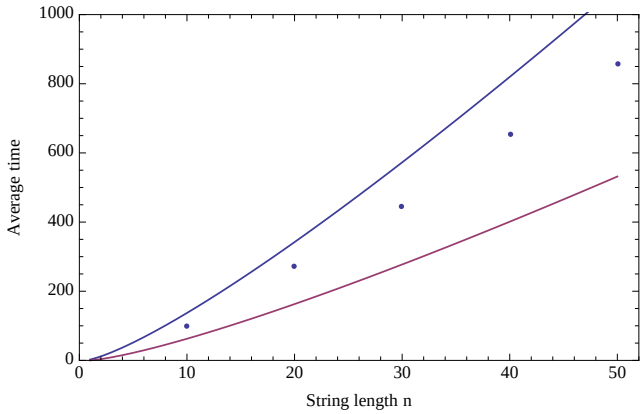

Figure 7: The average time for all islands to solve the OneMax problem on n bits, using a ring topology with $m = n$ islands, and migration probability $\mu = 0.1$. Each data point is averaged over 100 runs. Upper bound from derived result. Lower bound from a single copy of OneMax.

individuals can catch the disease (with some probability) if they come into contact with an infected individual. The simplest scheme assumes just these two states, and that when an individual recovers from a disease they become susceptible again. This is the so-called SIS model. Additional states can be introduced to represent more realistic features, such as *recovered*, in which an individual gains some immunity to the disease by virtue of having been infected. This gives the SIR model. Additionally, birth and death processes (both from the disease and from other causes) may be considered, and various schemes for controlling the disease, such as vaccinations.

The traditional analysis assumes that the population is perfectly mixed, meaning that an individual is equally likely to encounter any of the rest of the population. The analysis then proceeds by studying the dynamics of the proportion of infected individuals, p. We follow [14], but simplify to the SIS model.

Over a given time period, let r be the probability that an infected individual recovers, and s the probability that a susceptible individual catches the disease from an infected individual. We model the process as follows. Let

$$F(p) = (1 - r)p + sp(1 - p)$$

then we follow the dynamics as

$$p(t + 1) = F(p(t)).$$

Notice that this assumes the dynamics is deterministic. It should also be noted that we are considering discrete-time dynamics. It is easily shown that the fixed-points of this system are $p = 0$ and $p = 1 - r/s$. This latter fixed-point is only valid if $s > r$ (since otherwise it is negative). We then perform a stability

analysis:

$$\frac{dF}{dp} = 1 - r + s - 2sp.$$

So at the fixed-point $p = 0$, we have asymptotic stability if $s < r$. If $s > r$ then this fixed-point becomes unstable (the differential is greater than 1), and the other fixed-point becomes asymptotically stable. The condition $s = r$ thus represents a critical threshold between the disease disappearing from the population, and it remaining.

An immediate issue with this model if we consider the stochastic process is that the state in which all individuals are susceptible represents the absorbing state of the system regardless of the parameter settings, and so in the long-term the population will arrive in this state and remain there forever. One hopes, therefore, to show that the threshold represents a condition distinguishing between the rapid convergence to the absorbing state, and the infection remaining present for a "long" transient period of time (e.g. exponential in the size of the population) before absorption.

We also, of course, need to address the assumption that the population is perfectly mixed, and this leads us to consider the process occuring on a network (or graph) in which nodes represent individuals, and edges describe which individuals may interact with each other. There is a considerable literature on such network models, and a good recent review can be found in [13]. Analysis of such systems usually involves making approximations such as mean-field assumptions. Exact results exist in a limited number of cases, for example when the graph is complete or has the star topology, in the context of continuous time updates (see [6], which also gives loose bounds for general network topologies).

We will consider the SIS process on a graph, using a synchronous update rule, as with cellular automata (see [5] — a common alternative is to assume that individuals update their state according to a Poisson process). Let $x_1, \ldots, x_n \in \{0, 1\}$ represent the state of the n individuals in the population, where 0 indicates *susceptible* and 1 indicates *infected*. Over a single time step, individuals update their states simultaneously according to the following rule:

If $x_i(t) = 1$ then $x_i(t+1) = 0$ with probability r, else $x_i(t+1) = 1$.

If $x_i(t) = 0$ then select a neighbour j uniformly at random. If $x_j(t) = 1$ then $x_i(t+1) = 1$ with probability s. Otherwise $x_i(t+1) = 0$.

We will assume that there are no isolated individuals, since these cannot influence whether or not an epidemic takes hold in the network.

In this scheme, the deterministic dynamical equations are now seen to represent expectations:

$$E[x_i(t+1) \,|\, x_1(t), \ldots, x_n(t)] = (1 - r)x_i(t) + \frac{s(1 - x_i(t))}{\nu_i} \sum_{j \sim i} x_j(t)$$

where $\sim$ indicates the neighbour relation and ν_i is the degree of node i. We show that the condition $s < r$ is indeed sufficient to ensure the rapid disappearance of the disease.

Theorem 5. *In the synchronous SIS model on a graph with infection rate s and recovery rate r, if $s < r$ then the population will entirely recover from the disease in expected time bounded above by*

$$\frac{1 + \log(2|E|/\nu^*)}{r - s}$$

where E is the edge set of the graph and ν^* the smallest degree of the graph.

Proof
From the above description, it follows that

$$E[x_i(t+1) \,|\, x_1(t), \ldots, x_n(t)] \leq (1-r)x_i(t) + \frac{s}{\nu_i} \sum_{j \sim i} x_j(t).$$

This gives us:

$$A_{i,j} = \begin{cases} 1 - r & \text{if } i = j, \\ \frac{s}{\nu_i}[j \sim i] & \text{if } i \neq j. \end{cases}$$

It can be checked that $\nu = (\nu_1, \ldots, \nu_m)$ is a left eigenvector, with corresponding eigenvalue $\lambda = 1 - r + s$. The result then follows from the linear multi-objective drift theorem. □

It is not hard to imagine that the condition $s < r$ might not be necessary to ensure rapid absorption. Consider the ring topology, for example, with half (that is, $n/2$) of the nodes infected. If the infected nodes alternate with the susceptible nodes around the ring, then one expects $rn/2$ of them to recover, and $sn/2$ of the susceptible nodes to become infected. One can see that in this situation a balance is acheived when $r = s$. However, if the $n/2$ infected individuals lie contiguously on the ring, then again $rn/2$ will be expected to recover, but only s nodes in expectation will get infected (that is, either or both of the two nodes bordering the contiguous block of infected individuals). For these situations, a much stronger infection rate will be required to prevent progress towards the absorbing state.

To analyse the situation on a ring more closely, we first consider the count of how many infected individuals there are in the population. That is, let

$$Z(t) = \sum_i x_i(t).$$

Then we have

$$E[Z(t+1) \,|\, x_1(t), \ldots, x_n(t)] = (1-r)Z(t) + \frac{s}{2} \sum_i (1 - x_i) \sum_{j \sim i} x_j(t).$$

Now

$$\sum_i (1 - x_i) \sum_{j \sim i} x_j(t) = U + V$$

where

$$U = \sum_i (1 - x_i)x_{i-1}$$

and

$$V = \sum_i (1 - x_i)x_{i+1}$$

where we add and subtract modulo the number of nodes. However, we now see

$$\sum_i (1-x_i)x_{i-1} = \sum_i x_{i-1} - \sum_i x_{i-1}x_i = \sum_i x_{i+1} - \sum_i x_i x_{i+1} = \sum_i (1-x_i)x_{i+1}$$

and so in fact $U = V$. Therefore we have

$$E[Z(t+1) \,|\, x_1(t), \ldots, x_n(t)] = (1-r)Z(t) + sV(t).$$

Define the indicator function

$$\xi_i = [x_i = 0 \text{ and } x_{i+1} = 1]$$

so that $V = \sum_i \xi_i$. To determine the expected value of V at time $t+1$ we therefore examine the expectation that ξ_i takes the value 1 at this time, which is just the same as the probability that it takes that value. This, of course, depends on the previous time step and there are four cases to consider.

Case 1 Suppose at time t we have $x_i = 1$ and $x_{i+1} = 1$. Then the probability that $\xi_i = 1$ at the next time step is $r(1-r)$.

Case 2 Suppose at time t we have $x_i = 1$ and $x_{i+1} = 0$. Now not only does node i have to recover, but node $i+1$ has to become infected. The probability for these two events to occur is at most rs.

Case 3 Suppose at time t we have $x_i = 0$ and $x_{i+1} = 1$. Now node i must not become infected and node $i+1$ must not recover. The latter happens with probability $1-r$. The probability that i gets infected is at least $s/2$ (since it has at least one infected neighbour). So the chance that it is not infected is at most $1 - s/2$. Thus, the probability that $\xi = 1$ in this case is at most $(1-r)(1-s/2)$.

Case 4 Suppose at time t we have $x_i = 0$ and $x_{i+1} = 0$. We divide this into two sub-cases, depending on the value of x_{i+2}. For if $x_{i+2} = 0$, then it is impossible for node $i+1$ to become infected. If, however, $x_{i+2} = 1$, then it becomes infected with probability $s/2$. We also need node i to not be infected. All we can say about this is that this happens with probability at most 1. Thus the probability, in this case, of having $\xi_i = 1$ is bounded above by $[x_{i+2} = 1]s/2$.

Putting all these cases together, we have

$$\begin{aligned} E[\xi_i(t+1) \mid x_1, \ldots, x_n] &\leq [x_i = 0][x_{i+1} = 0][x_{i+2} = 1]s/2 \\ &+ [x_i = 0][x_{i+1} = 1](1-r)(1-s/2) \\ &+ [x_i = 1][x_{i+1} = 0]rs \\ &+ [x_i = 1][x_{i+1} = 1]r(1-r). \end{aligned}$$

Summing gives

$$\begin{aligned} E[V(t+1) \mid x_1, \ldots, x_n] &\leq (s/2) \sum_i [x_i = 0][x_{i+1} = 0][x_{i+2} = 1] \\ &+ (1-r)(1-s/2) \sum_i [x_i = 0][x_{i+1} = 1] \\ &+ rs \sum_i [x_i = 1][x_{i+1} = 0] \\ &+ r(1-r) \sum_i [x_i = 1][x_{i+1} = 1]. \end{aligned}$$

Taking these sums one at a time, we see

$$\begin{aligned} &\sum_i [x_i = 0][x_{i+1} = 0][x_{i+2} = 1] \leq \sum_i [x_{i+1} = 0][x_{i+2} = 1] = V. \\ &\sum_i [x_i = 0][x_{i+1} = 1] = V. \\ &\sum_i [x_i = 1][x_{i+1} = 0] = \sum_i [x_{i-1} = 1][x_i = 0] = U = V. \\ &\sum_i [x_i = 1][x_{i+1} = 1] = \sum_i [x_i = 1][x_{i+1} = 1] + \sum_i x_i - \sum_i x_i \\ = \quad &\sum_i [x_i = 1][x_{i+1} = 1] + \sum_i x_i - \sum_i x_{i+1} \\ = \quad &\sum_i x_i - \sum_i (1 - x_i)x_{i+1} = Z - V. \end{aligned}$$

So overall we have

$$E[V(t+1) \mid x_1, \ldots, x_n] \leq (s/2)V + (1-r)(1-s/2)V + rsV + r(1-r)(Z-V)$$

which gives:

$$E[V(t+1) \mid x_1, \ldots, x_n] \leq r(1-r)Z + (1 - 2r + 3rs/2 + r^2)V.$$

We now have two distance functions (Z and V) whose expected progress is bounded by a linear function. Applying the lemma for two distance functions, we see that the time to hit the state where all infection has disappeared (since both Z and V are zero) is logarithmic if

$$r(2r - 3rs/2 - r^2) > sr(1-r).$$

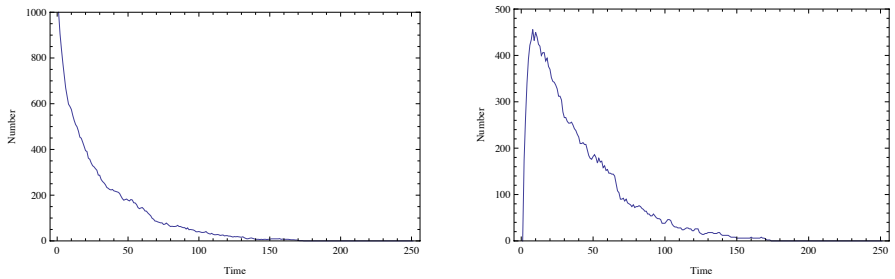

Figure 8: The number of infected nodes (left) and mixed edges (right) in a typical run on a ring of 1000 nodes, starting with all of them infected. $r = 0.1$ and $s = 0.15$.

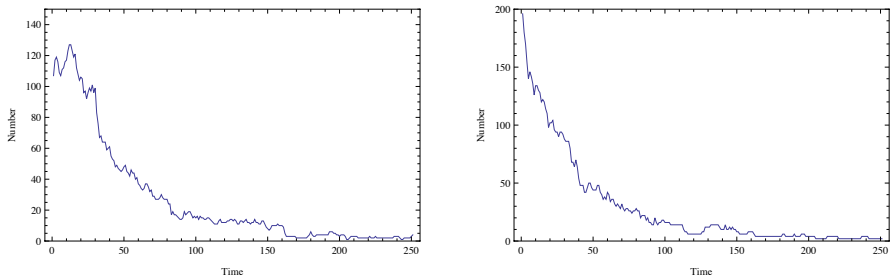

Figure 9: The number of infected nodes (left) and mixed edges (right) in a typical run on a ring of 1000 nodes, starting with 10% of them infected. $r = 0.1$ and $s = 0.15$.

Simplifying, this yields the condition

$$s < \frac{2r(2-r)}{2+r}.$$

We can see from this that a population with the ring topology can withstand much higher infection rates than a well-mixed population. For small values of r, the infection rate can be almost double the recovery rate, and the infection still disappear rapidly. We will now look at some experiments to investigate this further.

7. Experiments

The first experiment looks at the progress in each of two variables: the total number of infected nodes (Z), and the number of mixed edges ($U + V$). In Figure 8 a typical run is shown for a ring of 1000 nodes, starting with them all infected. When there are a large number of infected nodes, the number of mixed edges will be small (zero to begin with) and there will typically be a loss in infected nodes with a corresponding gain in mixed edges. After a while, the number of mixed-edges also begins to decrease, as the number of nodes becomes rather low, before the infection finally disappears.

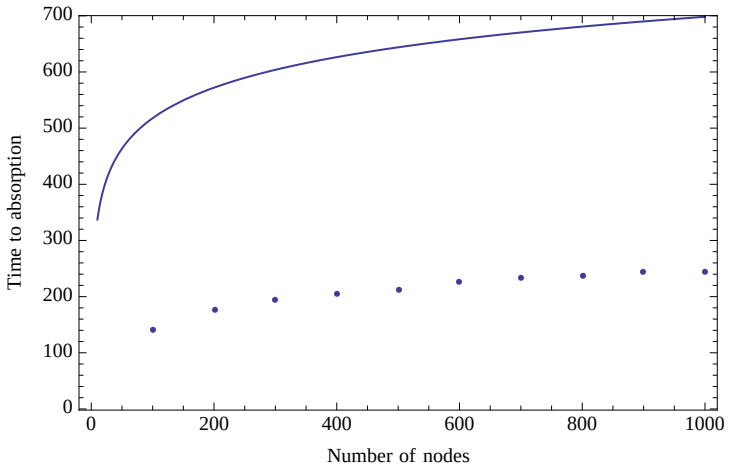

Figure 10: The average time to the elimination of the disease. Each data point is averaged over 100 runs. The solid line is the upper bound. $r = 0.1$ and $s = 0.15$.

In contrast, Figure 9 shows a run, again with 1000 nodes, but starting with a random 10% infected. Most of these infected nodes will be isolated, and with a strong chance of infecting their neighbours. We therefore see the number of infected nodes increasing initially, while the number of mixed edges decreases. Eventually, the infected nodes form blocks such that the further increase in infection is offset by the larger number recovering, and the infection disappears.

From these examples, it can be seen that the drift in each variable, considered separately, is not necessarily monotonic towards the final state. The direction of drift depends on the particular configuration at the time.

We now look at the time to the elimination of the disease, compared to the upper bound given theoretically. Experiments show that the upper bound considerably over-estimates the time required. Figure 10 shows the theoretical upper bound in the case of $r = 0.1$ and $s = 0.15$ across a range of population sizes, together with experimental data, each averaged over 100 runs. In each case, the start state was with every node infected. The constants in the upper bound were calculated from the eigenvalues and eigenvectors of the matrix containing the coefficients of the drift equations.

Of more interest is the prediction that the critical threshold for the infection rate causing the disease to remain in the population, is considerably higher than $s = r$. Our analysis shows that levels of s close to $2r$ still lead to the disease being eliminated rapidly. Given the worst-case bounds that were applied, one would expect the true threshold to be even higher.

To test this experimentally, we fixed a particular value of $r = 0.1$ and then examined the time to absorption setting s at various multiples of r, until a level was reached when the run did not terminate within a reasonable time. For this purpose, we used a fixed population size of 100 individuals, and again started with all of them infected. The results, shown in Figure 11 show the actual

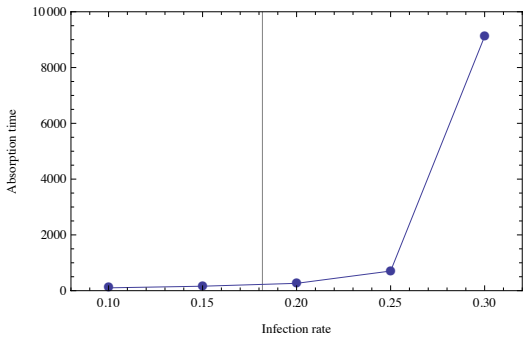

Figure 11: The average time to the elimination of the disease, for different infection rates. The ring has 100 nodes, and each data point is averaged over 100 runs (all initialy infected). The vertical line is the lower bound for the threshold. $r = 0.1$.

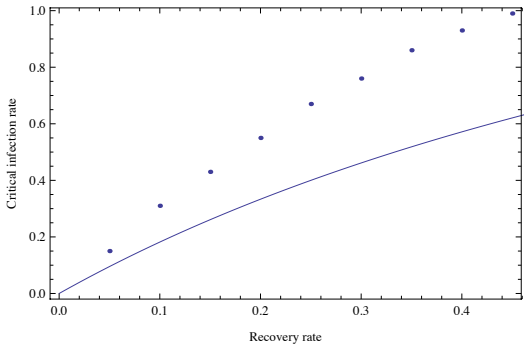

Figure 12: The empirically estimated critical infection rate (dots), for different recovery rates, compared to the theoretical lower bound of (solid line). Ring topology, 100 nodes.

threshold for s is somewhat higher than the lower-bound derived.

The critical threshold for the infection rate, for different recovery rates, was estimated experimentally by conducting a binary search on possible values to identify the largest value (to 2 decimal places) for which the infection disappeared within a reasonable time (for a 100 node ring, this was set to 50,000 time steps). The results, shown in Figure 12, illustrate these compared to the theoretical lower bound. It is of interest that for recovery rates $r > 0.45$, the infection dies out no matter how large the infection rate is.

The conservative nature of our bounds can be seen from these results, arising partly from the use of the multiplicative drift theorem, and partly because of the worst-case assumptions on the distribution of infected nodes in the case analysis.

8. Conclusions

We have presented a generalisation of the multiplicative drift theorem to the case where a process or algorithm is best described by more than one distance function. This allows the run-time analysis of situations of this sort — we have given examples from multi-objective evolutionary algorithms, parallel evolution on islands with migration and stochastic process on networks such as a simple epidemiology model.

The proposed tool is not strictly more powerful than multiplicative drift, in that it provides conditions under which a new single distance function with mutliplicative drift exists. However, the defining of suitable distance functions for particular problems can be a challenge and there are cases, as in our examples, where there may be several relevant natural distances, none of which in themselves follow the multiplicative drift condition. The result is particularly useful when there are two distance functions, where the relevant eigensystem is easy to compute explicitly. Problems with a larger number of objectives need sufficient structure for this to be achieved.

We have illiustrated three possible causes of how the multiplicative drift theorem can sometimes result in a large over-estimate of the run-time:

1. When large steps towards the target can occur.
2. When the variance in the run-time is large.
3. When the state closest to the target is unlikely to be visited.

The linear multi-objective drift theorem presented in this paper naturally inherits all these issues.

For single distance functions, multiplicative drift can be seen as a special case of variable drift [8, 12], where the expected progress is bounded by an arbitrary function (subject to minor conditions). It is hard to see how to generalise this case to multiple distance functions, without adding considerable extra assumptions. For example, the differential equation method, due to Wormald [15] can apply if the progress is known fairly exactly in each distance function, and where additionally the probability of making large jumps is small. Even with such extra assumptions, the error inherent in Wormald's theorem can make it hard to produce the run-time bounds which we would like to see [4]. Pursuing these lines of thought and resolving these issues would be a worthwhile line of research.

Acknowledgements. The author would like to thank the organisers and participants of the Schloss Dagstuhl seminar 17191 on "Theory of Randomised Search Heuristics" where these ideas were initially presented, and the anonymous reviewers who helped to improve the paper considerably.

References

[1] B. Doerr and L. A. Goldberg. Drift analysis with tail bounds. In R. Schaefer, C. Cotta, J. Kolodziej, and G. Rudolph, editors, *Parallel Problem Solv-*

ing from Nature, PPSN XI, volume 6238 of *Lecture Notes in Computer Science*, pages 174–183. Springer, 2010.

[2] B. Doerr, D. Johannsen, and C. Winzen. Multiplicative drift analysis. *Algorithmica*, 64(4):673–697, 2012.

[3] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.

[4] T. Friedrich, T. Kötzing, and A. Melnichenko. Analyzing search heuristics with differential equations. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 313–314, 2017.

[5] M. A. Fuentes and M. N. Kuperman. Cellular automata and epidemiological models with spatial dependence. *Physica A: Statistical Mechanics and its Applications*, 267(3–4):471–486, 1999.

[6] A. Ganesh, L. Massoulie, and D. Towsley. The effect of network topology on the spread of epidemics. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1455–1466, 2005.

[7] I. Hanski and O. E. Gaggiotti. *Ecology, Genetics and the Evolution of Metapopulations*. Elsevier Academic Press, 2004.

[8] D. Johannsen. *Random combinatorial structures and randomized search heuristics.* PhD thesis, International Max Planck Research School for Computer Science, Universität des Saarlandes, 2010.

[9] P. K. Lehre and C. Witt. General drift analysis with tail bounds. *CoRR*, abs/1307.2559, 2013.

[10] P. K. Lehre and C. Witt. Concentrated hitting times of randomized search heuristics with variable drift. In H. K. Ahn and C. S. Shin, editors, *Algorithms and Computation. ISAAC 2014.*, volume 8889 of *Lecture Notes in Computer Science*, pages 686–697. Springer, 2014.

[11] T. M. Liggett. *Interacting Particle Systems*. Springer, 2005.

[12] B. Mitavskiy, J. E. Rowe, and C. Cannings. Theoretical analysis of local search strategies to optimize network communication subject to preserving the total number of links. *International Journal of Intelligent Computing and Cybernetics*, 2(2):243–284, 2009.

[13] R. Pastor-Satorras, C. Castellano, P. V. Mieghem, and A. Vespignani. Epidemic processes in complex networks. *Reviews of Modern Physics*, 87(3):925–979, 2015.

[14] J. H. Vandermeer and D. E. Goldberg. *Population ecology*. Princeton University Press, 2003.

[15] N. C. Wormald. The differential equation method for random graph processes and greedy algorithms. In M. Karoński and H. Prömel, editors, *Lectures on Approximation and Randomized Algorithms*, pages 73–155. PWN, Warsaw, 1999.